Toward a Common Data Model for Supercomputing

Livermore members of the award-winning HDF5 team are (from left) Robb Matzke, Mark Miller, Linnea Cook, and Kim Yates.

LAWRENCE Livermore won not five but six R&D 100 Awards in 2002. The October 2002 issue of this magazine described five awards for which Livermore was the primary developer. But the Laboratory was also a key member of a four-institution team that won a sixth R&D 100 Award for a software project called Hierarchical Data Format 5 (HDF5). As the primary developer, the National Center for Supercomputing Applications (NCSA) at the University of Illinois (Urbana–Champaign) worked in collaboration with Livermore, Los Alamos, and Sandia national laboratories.

HDF5 is the first widely used input/output (I/O) library specifically designed for massively parallel computing systems. An I/O library is a collection of programming routines that a computing code calls on to write or read data to or from a file on disk.

In the early years of scientific computing, roughly 1959 to 1980, I/O libraries took the form of little more than Fortran format statements to write or read binary or ASCII (text) files. Between 1980 and 2000, an important innovation emerged—the general purpose I/O library, which has two variants working at slightly different levels of data abstraction. One focused on computer science data structures, the other on computational science mesh structures. In addition to providing random access to named, abstract data objects, these libraries provided data that could be written on one machine architecture and read on another.

With the advent of massively parallel computers, computer scientists identified the need for a general-purpose I/O library that writes data from multiple processors into a single file, that can handle individual data structures larger than a gigabyte and individual files larger than a terabyte, and whose performance scales to large numbers of processors.

"With HDF5, we were also aiming at another big goal—to lay the foundation for solving the interoperability problem in scientific computing," says Livermore computer scientist Linnea Cook. "Many of the major supercomputing centers around the world—including Livermore, Los Alamos, and Sandia—had developed their own I/O libraries to address requirements not satisfied by existing libraries. None of these institutions could easily share data or software tools."

Mark Miller, a Livermore expert on data modeling, agrees. "With HDF5, we have the beginnings of a solution to the interoperability problem. Plus we made an important contribution to the scientific community."

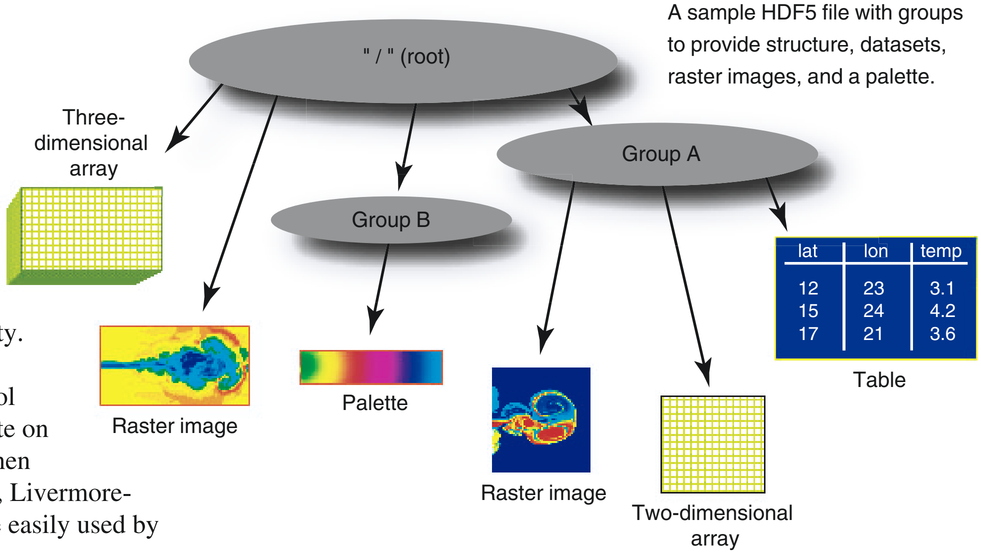
Problem, Solution

HDF5 is the successor to HDF1 through HDF4, a series of highly successful I/O libraries developed at NCSA and used around the world. HDF software includes I/O tools for analyzing, visualizing, and converting data between various numeric formats and storage schemes. By the mid-1990s, however, HDF4 had begun to show its limitations. It lacked the capacity to handle the enormous amounts of data being generated by many scientific research programs, and it was not designed to support parallel applications.

At about the same time, the Department of Energy established its Accelerated Strategic Computing Initiative (ASCI), now the National Nuclear Security Administration's Advanced Simulation and Computing Program, to develop a massively parallel computing capability. Livermore, Los Alamos, and Sandia established a group to explore the possibilities of developing visualization codes jointly. Cook served on this committee and initiated the effort to develop a common, tri-laboratory, parallel I/O library. She believed that the HDF group at NCSA might be part of the solution and invited them to participate.

Throughout the late 1990s, the three national laboratories worked with NCSA under ASCI auspices to develop an improved version of HDF. Livermore computer scientist Robb Matzke developed the majority of the first production versions of HDF5, with colleague Kim Yates making substantial contributions to the parallel version.

The team's primary goal was to produce a high-performance, parallel I/O library that would meet the requirements of all three laboratories. This meant that HDF5 needed to be delivered quickly, with sufficient functionality, robustness, and performance, before other libraries being built at all three laboratories became entrenched. The second goal was to develop a de facto standard library that would be widely used throughout the international scientific computing community. Once an I/O library becomes widely used, commercial software companies and free tool developers begin developing codes to operate on data stored in that format. Many users can then quickly and easily use these tools. Likewise, Livermore-built tools that read HDF5 data will be more easily used by other organizations that use HDF5.



lat	lon	temp
12	23	3.1
15	24	4.2
17	21	3.6

A sample HDF5 file with groups to provide structure, datasets, raster images, and a palette.

Today, HDF5 can store, access, manage, exchange, and archive not only massive amounts of complex data but also any type of data suitable for digital storage, no matter its origin or size. HDF5 is a completely portable file format. A file can be written on any system and read on any other. Plus its I/O libraries can run on virtually any scientific research computing system, be it serial or parallel. The HDF5 file format and library are designed to evolve as requirements emerge.

HDF5 can handle, for example, trillions of bytes of computational modeling data or high-resolution electronic images in a continuously evolving computing and storage environment. With the help of lower-level libraries, HDF5 enables thousands of processors to simultaneously write data to a single file. Terabytes of remote-sensing data received from satellites, computational results from weather or nuclear testing models, and high-resolution magnetic resonance imaging brain scans can be stored in HDF5 files along with additional information needed for efficient data exchange, data processing, visualization, and archiving.

HDF5 can also write data to a file on disk, to memory, across the network, or to any device specified by using its Virtual File Layer. Applications can write a virtually unlimited number of objects to a file, and the maximum size of any object is limited only by the computer or the file system's capacity. HDF5 provides simple data types, user-defined data types, and compound data types of any degree of complexity, including nesting to any number of levels. Through its grouping and linking mechanism, the HDF5 data model supports complex data relationships and dependencies. Multidimensional arrays can be extended in any dimension. In addition, HDF5 provides partial I/O capabilities. For example, users may wish to read only a subset of an array containing hundreds of millions of elements.

Work continues on HDF5—as well as on the broader interoperability problem it helps to address. Groups that collaborated on this award-winning technology are also working on higher level libraries to capture the mathematical and physical abstractions of scientific data. In addition, the HDF group is defining conventions for using HDF5, which will also improve interoperability.

Some Interesting Uses for HDF5

HDF5 is used by government, academic, and commercial institutions in more than 60 countries. The three DOE laboratories are currently the largest users of HDF5's parallel capabilities, but that is beginning to change.

HDF5 is incorporated into the Globus Project at Argonne National Laboratory, which focuses on the fundamental technologies required to deploy computational grids. Argonne's NeXus, which provides a standard data format for work in neutron and synchrotron radiation internationally, is converting to HDF5. FLASH, a product used to study thermonuclear flashes on the surfaces of compact stars, is built on HDF5.

The big surprise of 2001 for the HDF group came when the Help Desk started getting e-mail messages from a programmer in New Zealand. His questions were not of the ordinary sort, and the technical support staff eventually asked him how he was using the library. All he was free to say at the time was that he was working on graphical special effects. Later information revealed that his company was using HDF5 to generate atmospheric effects—smoke, wind, and clouds—for *The Lord of the Rings* film trilogy.

—Katie Walter

Key Words: Advanced Simulation and Computing (ASCI), Hierarchical Data Format 5 (HDF5), input/output (I/O) libraries, R&D 100 Award, supercomputing.

For further information contact Linnea Cook (925) 422-1686 (cook13@llnl.gov).